UC San Diego


Analyzing Generalization of Neural Networks through Loss Path Kernels

Y. Chen[1], W. Huang[2], H. Wang[3], C. Loh[3], A. Srivastava[3], L. Nguyen[4], and T.-W. Weng[1]

[1]UCSD, [2]RIEKN AIP, [3]MIT-IBM Watson AI Lab, [4]IBM Research

03/04/2024

Outline

1. Introduction and motivation
2. Main results
3. Conclusion and future works

Outline

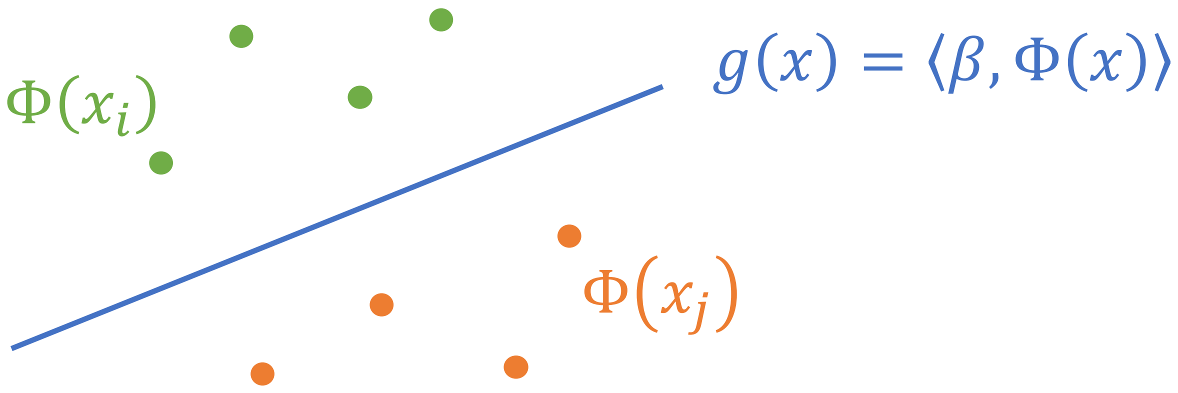
1. Introduction and motivation
   - Kernel machine and neural tangent kernel
   - Generalization theory of neural networks
   - Motivation of this work
2. Main results
   - Loss path kernel and the equivalence between NN and KM
   - Generalization bound for NN trained by gradient flow
   - Case study and Application
     - Ultra-wide NN
     - Neural architecture search
3. Conclusion and future works

Kernel Machine

- Kernel: $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$, Φ: $\mathcal{X} \rightarrow \mathcal{H}$ maps the data to a (potentially infinite dimensional) feature space.
- Kernel machine (KM): linear function in the feature space

$$g(x) = \langle \beta, \Phi(x) \rangle + b = \sum_{i=1}^{n} a_i K(x, x_i) + b, \quad \text{where } \beta = \sum_{i=1}^{n} a_i \Phi(x_i)$$

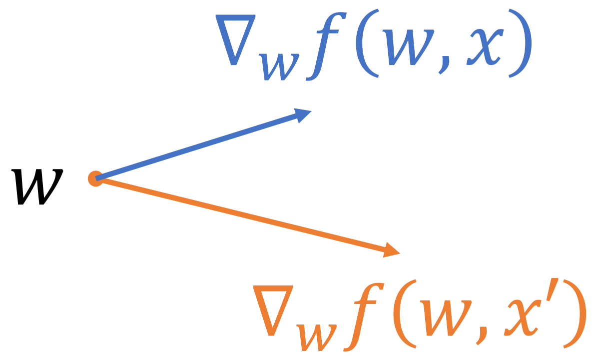
- RKHS norm of g: $\|\beta\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j K(x_i, x_j)}$



Neural tangent kernel

- Neural Tangent Kernel (NTK) (Jacot et al., 2018):

$$\widehat{\Theta}(w;\ x, x') = \langle \nabla_w f(w, x), \nabla_w f(w, x') \rangle$$



measures the similarity between data points x, x' by comparing their gradients

- Under certain conditions (e.g., infinite width limit and NTK initialization), NTK at initialization w_0 converges to a deterministic limit and keeps constant during training:

$$\widehat{\Theta}(w_0;\ x, x') \to \Theta_\infty(x, x')$$

NTK at initialization   Independent of w_0

Neural tangent kernel

- Infinite-width neural network (NN) trained by gradient descent with mean square loss ⇔ kernel regression with NTK [Jacot et al., 2018; Arora et al., 2019]

kernel regression: linear regression in the feature space

$$\min_{w_t} \sum_{i=1}^{n} \|\langle w_t - w_0 , \nabla_w f(w_0, x_i)\rangle - y_i\|^2$$

- Wide NNs are linear in the parameter space [Lee et al., 2019]:

$$f(w_t, x) = f(w_0, x) + \langle \nabla_w f(w_0, x), w_t - w_0\rangle + O(\frac{1}{\sqrt{m}})$$

m: width of NN

- Infinite-width NN trained with ℓ_2 regularized loss ⇔ ℓ_2 regularized KMs with NTK, e.g. SVM [Chen et al., 2021]

Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Neural tangent kernel

Ultra-wide NN trained with

equivalent

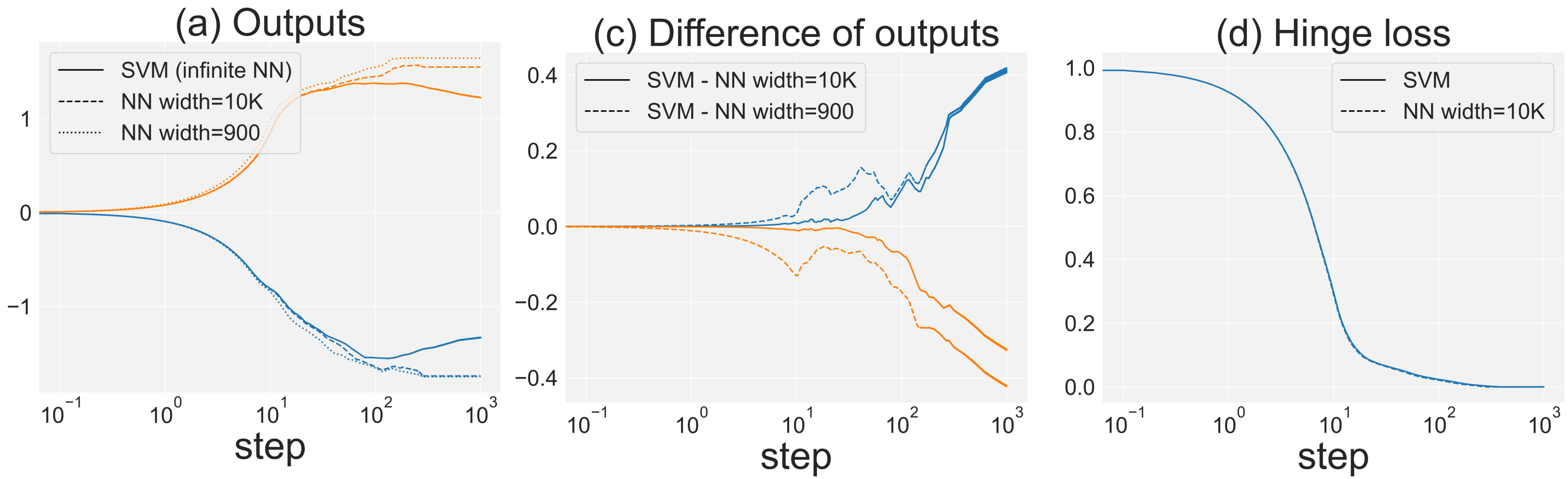
KM $g(\beta, x) = \langle \beta, \nabla_w f(w_0, x) \rangle$ with

$$L(w) = \frac{\lambda}{2} \left\| w^{(L)} \right\|^2 + \sum_{i=1}^{n} \ell(f(w, x_i), y_i)$$

$$L(\beta) = \frac{\lambda}{2} \|\beta\|^2 + \sum_{i=1}^{n} \ell(g(\beta, x_i), y_i)$$

$w^{(L)}$: last layer of NN

SVM: $\ell(z, y) = \max(0, 1 - zy)$



Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Neural tangent kernel

Our prior work [Chen et al., 2021]:

λ	Loss $\ell(z, y)$	Kernel machine
$\lambda = 0$ ([1,2])	$(y - z)^2$	Kernel regression
$\lambda \to 0$ (ours)	$\max(0, 1 - yz)$	Hard margin SVM
$\lambda > 0$ (ours)	$\max(0, 1 - yz)$	(1-norm) soft margin SVM
	$\max(0, 1 - yz)^2$	2-norm soft margin SVM
	$\max(0, \lvert y - z\rvert - \epsilon)$	Support vector regression
	$(y - z)^2$	Kernel ridge regression (KRR)
	$\log(1 + e^{-yz})$	Logistic regression with ℓ_2 regularization

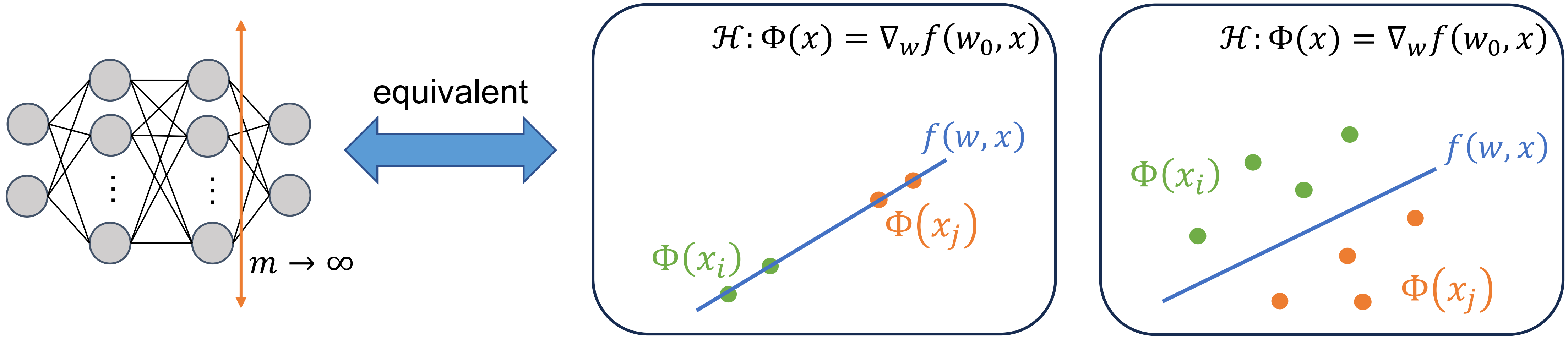
Suppose ℓ is Lipschitz and smooth,

$$\|f(x) - g(x)\| \leq \tilde{O}\left(\frac{1}{\sqrt{m}}\right)$$

Chen et al., 2021. On the equivalence between neural network and support vector machine. NeurIPS 2021.

Neural tangent kernel

kernel regression | ℓ_2 regularized KMs



[Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019]

[Chen et al., 2021]

These equivalences are useful for analyzing NNs

☹ But only holds for infinite-width/ultra-wide NNs

Q1. Can we establish a connection or equivalence between general NNs (vs ultra-wide NNs) and KMs?

Generalization theory of neural networks

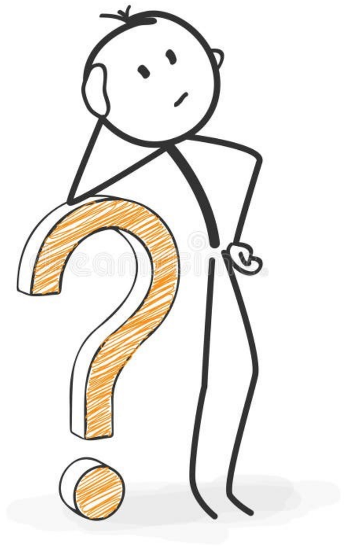
How do the neural networks (NN) generalize on test data?

generalization gap:

$$GAP = \underbrace{\mathbb{E}_{z\sim\mu}\left[\ell(w,z)\right]}_{L_\mu(w)\text{: population loss}} - \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell(w,z)}_{L_S(w)\text{: training loss}} \leq \boxed{?}$$

$$GAP \leq \sqrt{\frac{|\mathcal{G}|}{n}}$$

$\mathcal{G}$: NN function class
n: # of samples



Generalization theory: general NNs

1. VC dimension [Bartlett et al., 2019]

$$GAP \leq O\left(\sqrt{L\frac{\#\,of\,parameters}{n}}\log(n)\right)$$

L: # of layers
n: # of samples
W_l: weight of layer l

2. Norm-based bounds [Bartlett et al., 2017; …]

$$GAP \leq O\left(\frac{\prod_{l=1}^{L}\|W_l\|}{\sqrt{n}}\right)$$

☹

- Do not explain the generalization ability of overparameterized NNs. [Belkin et al., 2019]
- Vacuous: too large to be useful

- Other bounds:
  - PAC-Bayes bounds (mainly focus on stochastic NNs)
  - Information-theoretical approach (expected bound)

Bartlett, et al.. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. JMLR 2019.
Bartlett, et al.. Spectrally-normalized margin bounds for neural networks. NeurIPS 2017.

Generalization theory: ultra-wide NNs

- Arora et al., 2019: for ultra-wide two-layer FCNN,

$$GAP \leq \sqrt{\frac{2\,\mathbf{y}^{\top}(\mathbf{H}^{\infty})^{-1}\mathbf{y}}{n}}$$

$\mathbf{H}^{\infty}$: NTK of the first layer

- Cao & Gu, 2019: for ultra-wide L-layer FCNN,

$$GAP \leq \tilde{O}(L \cdot \sqrt{\frac{2\,\mathbf{y}^{\top}(\Theta)^{-1}\mathbf{y}}{n}})$$

☹ These bounds only hold for ultra-wide NNs

Q2. Can we establish tight (vs vacuous) generalization bounds for general NNs (vs ultra-wide NNs)?

Motivation of this work

1. Can we establish a connection or equivalence between general NNs (vs ultra-wide NNs) and Kernel machines? It can have many benefits:
    1. New understanding of NN trained with SGD
    2. Generalization bound for NNs from the perspective of kernel
    3. Analyze NN architectures from this equivalence
    4. Improve kernel method from the NN viewpoint

?

2. Can we establish tight (vs vacuous) generalization bounds for general NNs (vs ultra-wide NNs)?

Yes!

Outline

1. Introduction and motivation
    - Kernel machine and neural tangent kernel
    - Generalization theory of neural networks
    - Motivation of this work
2. Main results
    - Loss path kernel and the equivalence between NN and KM
    - Generalization bound for NN trained by gradient flow
    - Case study and Application
        - Ultra-wide NN
        - Neural architecture search
3. Conclusion and future works

Loss Path Kernel

Neural tangent kernel (NTK):

$$\widehat{\Theta}(w;\ x, x') = \langle \nabla_w f(w, x), \nabla_w f(w, x') \rangle$$

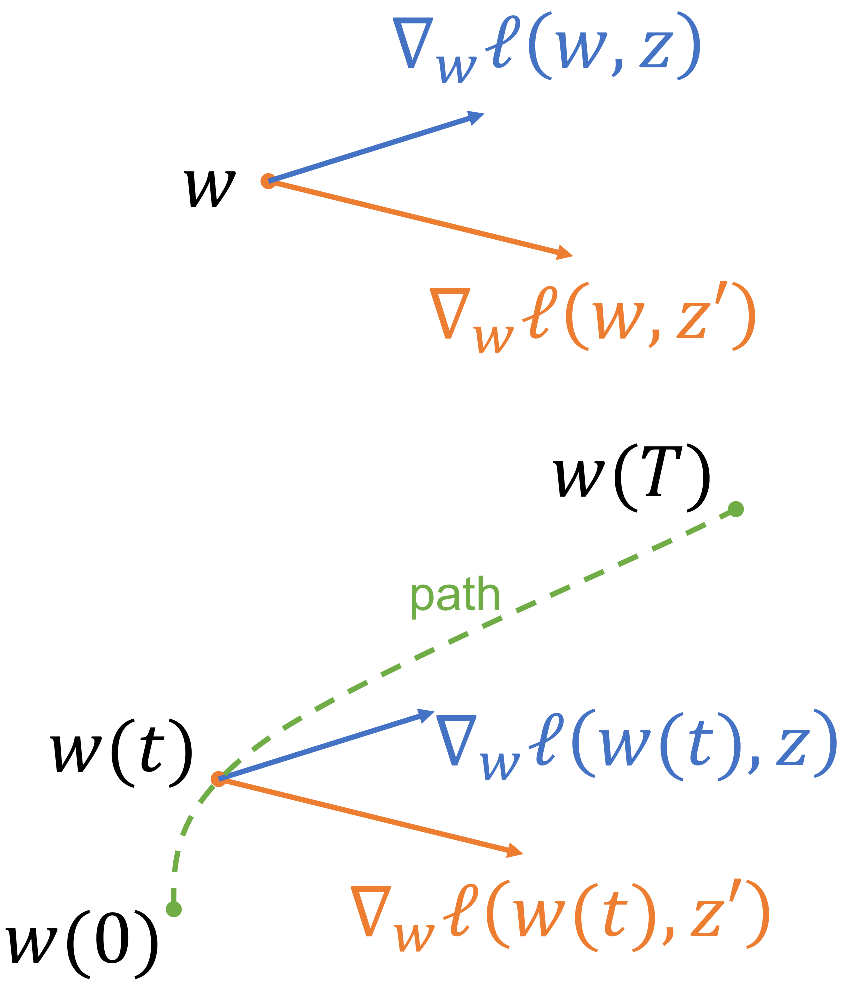
Loss Tangent Kernel (LTK): $z = (x, y)$

$$\overline{\mathrm{K}}(w; z, z') = \langle \nabla_w \ell(w, z), \nabla_w \ell(w, z') \rangle$$

Loss Path Kernel (LPK):

$$\mathrm{K}_T(z, z'; S) = \int_0^T \overline{\mathrm{K}}(w(t); z, z')\ dt$$

$$= \int_0^T \langle \nabla_w \ell(w, z), \nabla_w \ell(w, z') \rangle\, dt$$



Equivalence between neural network and kernel machine

With gradient flow (gradient descent with infinitesimal step size):

$$\frac{w(t+1) - w(t)}{\eta} = -\nabla_w L_S(w(t)) \xrightarrow{\eta \to 0} \frac{dw(t)}{dt} = -\nabla_w L_S(w(t))$$

We can derive equivalence:

$$\ell(w_T, z) = \boxed{\sum_{i=1}^{n} -\frac{1}{n} \mathrm{K}_T(z, z_i; S)} + \ell(w_0, z)$$

Loss function at time T

Kernel machine with **LPK**

Loss function at initialization

Very general equivalence!

Equivalence between neural network and kernel machine

Proof:

$$\frac{d\ell(w_t,z)}{dt} = \left\langle \nabla_w \ell(w_t,z), \frac{dw_t}{dt} \right\rangle$$ By chain rule

$$= \langle \nabla_w \ell(w_t,z), -\nabla_w L_S(w_t) \rangle$$ Gradient flow: $\frac{dw_t}{dt} = -\nabla_w L_S(w_t)$

$$= \left\langle \nabla_w \ell(w_t,z), -\frac{1}{n}\sum_{i=1}^{n} \nabla_w \ell(w_t,z_i) \right\rangle$$ Definition of $L_S(w_t)$

$$= -\frac{1}{n}\sum_{i=1}^{n} \langle \nabla_w \ell(w_t,z), \nabla_w \ell(w_t,z_i) \rangle$$ Rearrange

Integrate from 0 to T:

$$\ell(w_T,z) - \ell(w_0,z) = -\frac{1}{n}\sum_{i=1}^{n}\int_0^T \langle \nabla_w \ell(w_t,z), \nabla_w \ell(w_t,z_i) \rangle\, dt = \sum_{i=1}^{n} -\frac{1}{n} \mathrm{K}_T(z,z_i;S)$$

Equivalence between neural network and kernel machine

Stochastic gradient flow (SGD with infinitesimal step size):

$$\frac{w(t+1)-w(t)}{\eta} = -\nabla_w L_{S_t}(w(t)) \xrightarrow{\eta \to 0} \frac{dw(t)}{dt} = -\nabla_w L_{S_t}(w(t))$$

$S_t \subseteq \{1, \ldots, n\}$ is the indices of batch data, $m = |S_t|$: batch size

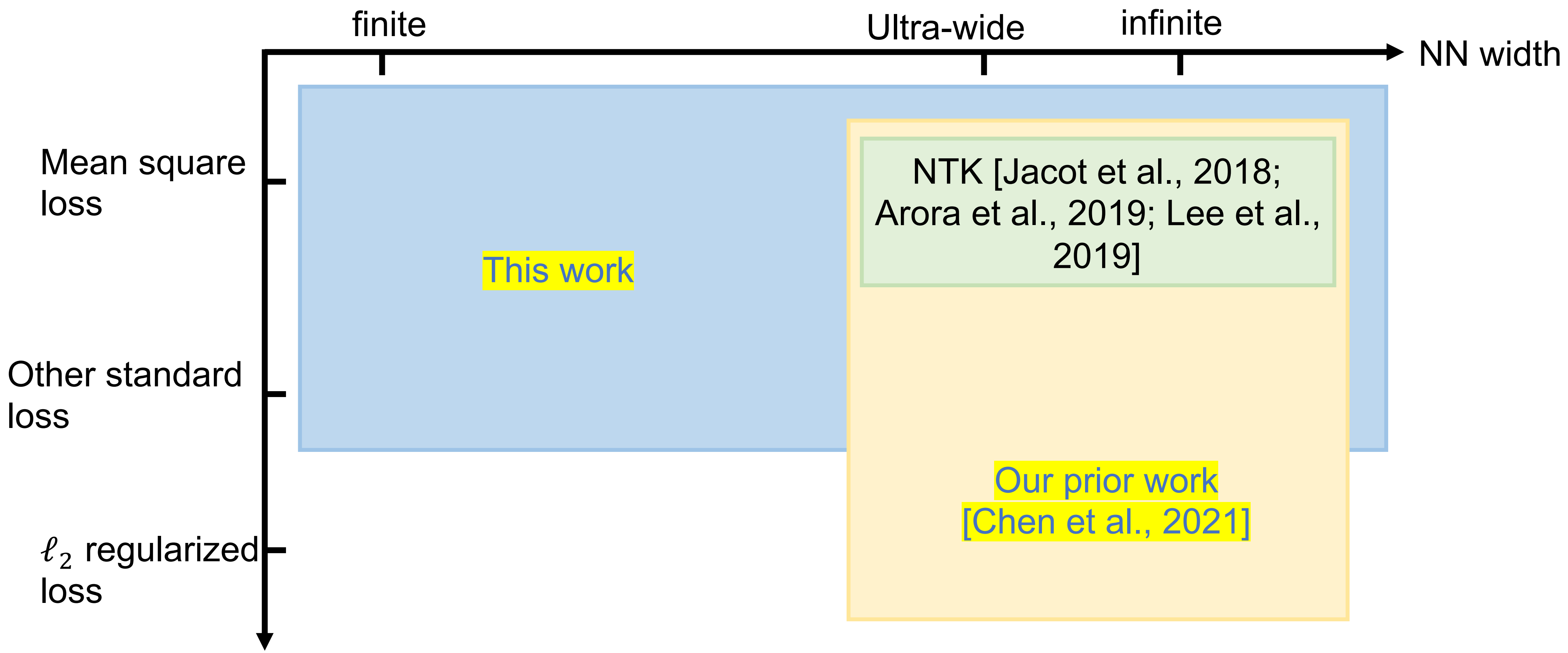
Equivalence:

Sum of KMs with **LPK**

$$\ell(w_T, z) = \boxed{\sum_{t=1}^{T-1} \sum_{i \in S_t} -\frac{1}{m} \mathrm{K}_T(z, z_i; S)} + \ell(w_0, z)$$

Equivalence between neural network and kernel machine

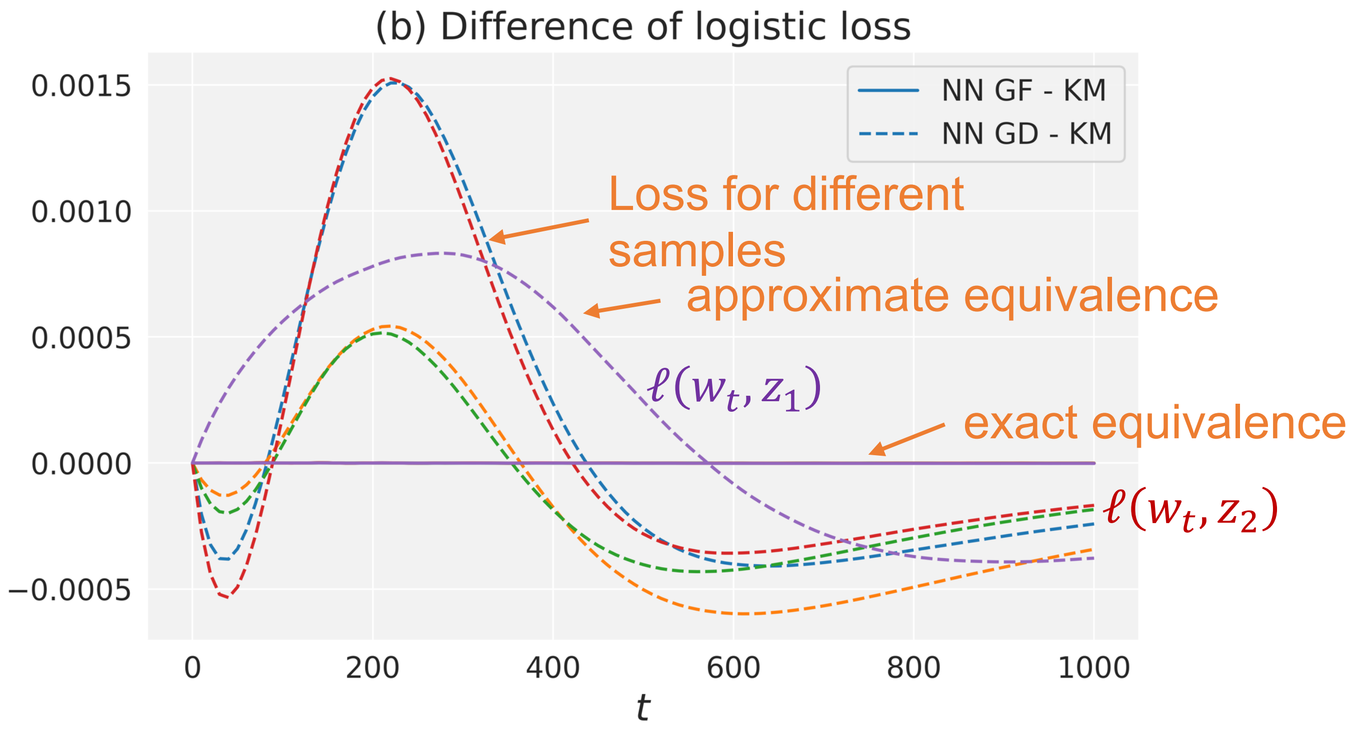
Compare with previous equivalence results:



Equivalence between neural network and kernel machine

Verify the equivalence: two-layer NN

$$\ell(w_T, z) - \left(\sum_{i=1}^{n} -\frac{1}{n}\mathbf{K}_T(z, z_i; S) + \ell(w_0, z)\right)$$



- NN trained by gradient flow (GF) exactly equal to the KM
- NN trained by gradient descent (GD) is also close with the KM

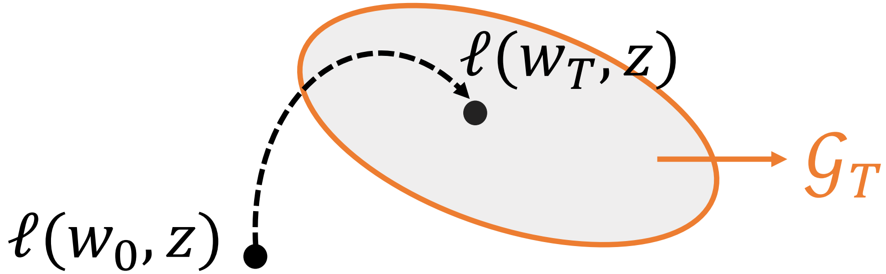
Generalization bound for NN trained by gradient flow

Different training set induces distinct LPK. Set of LPKs with constrained RKHS norm:

Set of LPKs

$$\mathcal{K}_T = \left\{\mathrm{K}_T(\cdot,\cdot;S'): S' \in \mathrm{supp}(\mu^{\otimes n}), \frac{1}{n^2}\sum_{i,j} \mathrm{K}_T\left(z_i', z_j'; S'\right) \leq B^2\right\}$$

$$S = \{z_i\}_{i=1}^n, \quad S' = \{z_i'\}_{i=1}^n$$



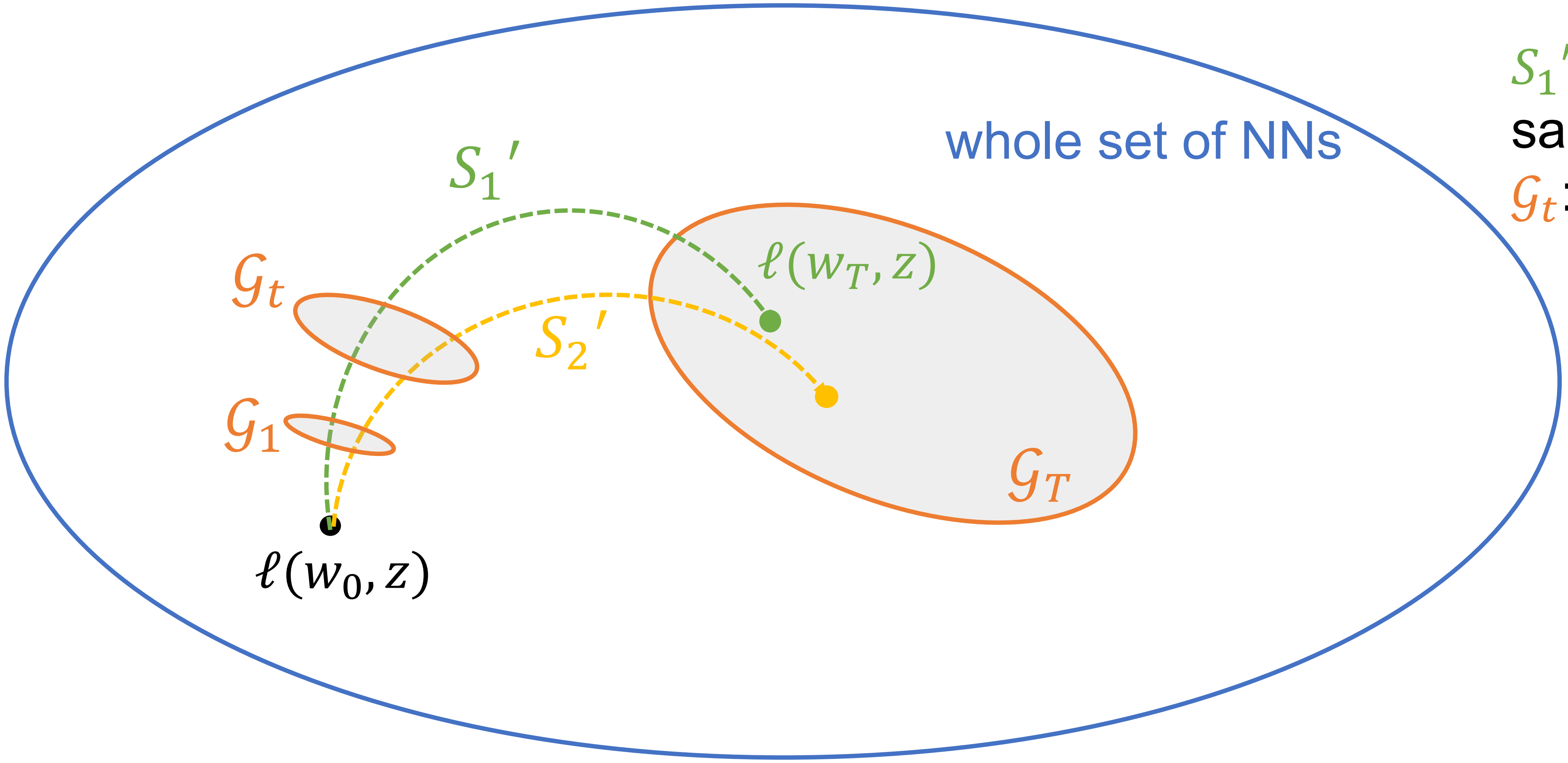
Set of NNs trained to time T from all feasible S' :

$$\mathcal{G}_T = \left\{ g(z) = \sum_{i=1}^{n} -\frac{1}{n}\mathrm{K}(z, z_i'; S') + \ell(w_0, z) : \mathrm{K}(\cdot,\cdot; S') \in \mathcal{K}_T \right\}$$

$\ell(w_T, z)$ trained from S'

Intuition of our work

- The set of trained NNs $\mathcal{G}_T$ can be much smaller than the whole set of NNs
- We characterize $\mathcal{G}_T$ through the equivalence between NN and KM



$S_1{}'$, $S_2{}'$: different set of samples

$\mathcal{G}_t$: set of NNs at time t

Rademacher complexity

Empirical Rademacher complexity of a function class $\mathcal{G}$,

$$\hat{\mathcal{R}}_S(\mathcal{G}) = \frac{1}{n}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}}\sum_{i=1}^{n}\sigma_i g(z_i)\right] \qquad \boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n) \sim \mathrm{Unif}(\{+1, -1\})$$

A measure of the richness of a function class. Measure the ability of the functions in $\mathcal{G}$ to correlate with random noise.

For $\mathcal{G} = \{\ell(f(x), y)\colon f \in \mathcal{F}\}$ and $\ell(f, y) \in [0, 1]$, with high probability [Mohri et al. 2018],

$$GAP \leq 2\hat{\mathcal{R}}_S(\mathcal{G})$$

Mohri, M., Rostamizadeh, A., and Talwalkar, A. Foundations of machine learning. MIT press, 2018.

Generalization bound for NN trained by gradient flow

Compute the Rademacher complexity of $\mathcal{G}_T$,

$$GAP \leq 2\min(U_1, U_2)$$

$$U_1 = \frac{B}{n}\sqrt{\sup_{\mathrm{K}\in\mathcal{K}_T}\sum_{i=1}^{n}\mathrm{K}(z_i, z_i; S') + \sum_{i\neq j}\Delta(z_i, z_j)}$$

$$\Delta(z_i, z_j) = \frac{1}{2}\left[\sup_{\mathrm{K}\in\mathcal{K}_T}\mathrm{K}(z_i, z_j; S') - \inf_{\mathrm{K}\in\mathcal{K}_T}\mathrm{K}(z_i, z_j; S')\right]$$

maximum magnitude of the loss gradient in $\mathcal{K}_T$ evaluated with S throughout the training trajectory.

$$\mathrm{K}_T(z_i, z_i; S') = \int_0^T \|\nabla_w \ell(w, z_i)\|^2\, dt$$

range of variation of LPK in $\mathcal{K}_T$

Can be estimated with training samples

Generalization bound for NN trained by gradient flow

Bound for general NNs

$$U_1 = \frac{B}{n}\sqrt{\sup_{\mathrm{K}\in\mathcal{K}_T}\sum_{i=1}^{n}\mathrm{K}(z_i, z_i; S') + \sum_{i\neq j}\Delta(z_i, z_j)}$$

Similar with the bound of KM but with an additional supremum over $\mathcal{K}_T$

Due to the set of kernels $\mathcal{K}_T$

- When $|\mathcal{K}_T| = 1$, U_1 recovers KM's bound

Bound for KM with a fixed kernel K

$$GAP \leq \frac{B}{n}\sqrt{\sum_{i=1}^{n} K(x_i, x_i)}$$

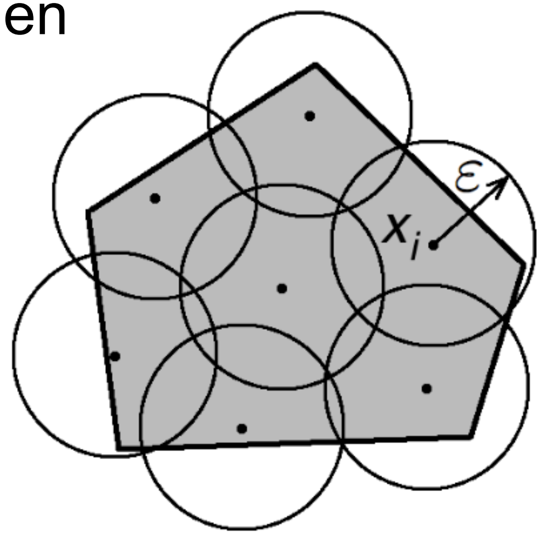
[Bartlett, P. L. and Mendelson, S. 2002]

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 2002.

Covering number

ϵ–cover: given a set U, distance ϵ, norm $||\cdot||$, $\mathrm{V} \subseteq U$ is a ϵ–cover of U when

$$\sup_{a\in U} \inf_{b\in V} ||a-b|| \leq \epsilon$$



Discretize or cover the function class with finite representative elements.

Covering number $\mathcal{N}(U, \epsilon, || \, ||)$: minimum cardinality of ϵ–cover.

- Can be used to analyze generalization
- Has a relation with Rademacher complexity

Generalization bound for NN trained by gradient flow

Analyze the covering number of $\mathcal{G}_T$,

$$GAP \leq 2\min(U_1, U_2)$$

$$U_2 = \inf_{\epsilon>0}\left(\frac{\epsilon}{n} + \sqrt{\frac{2\ln \mathcal{N}(\mathcal{G}_T^S, \epsilon, \|\ \|_1)}{n}}\right)$$

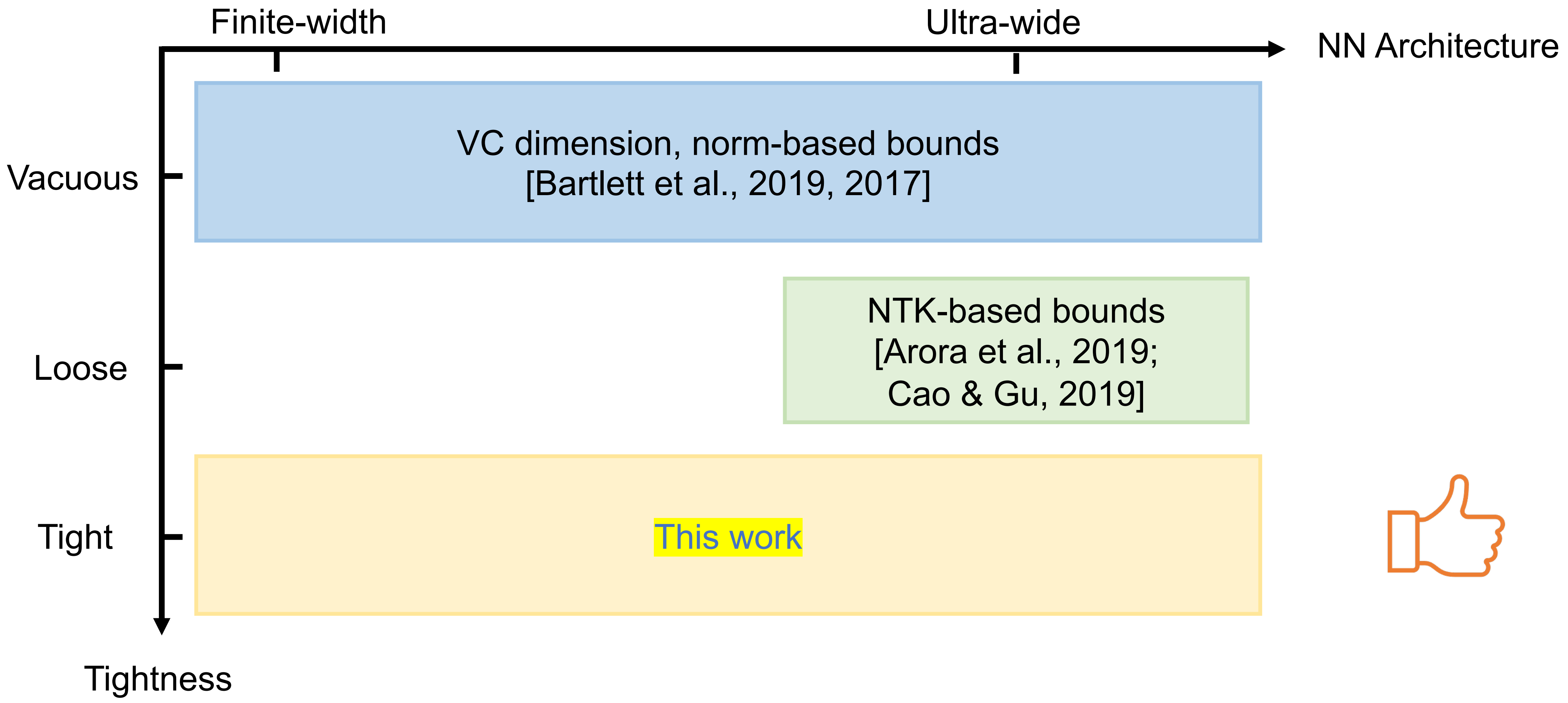
$\mathcal{G}_T^S = \{g(\mathbf{Z}) = \big(g(z_1), \dots, g(z_n)\big): g \in \mathcal{G}_T\}$,
$\mathcal{N}(\mathcal{G}_T^S, \epsilon, \|\ \|_1)$ is the covering number of $\mathcal{G}_T^S$.

If the variation of the loss with different training data is small,
U_2 will be small.

- Can be estimated with training samples
- Can get similar bounds as U_1, U_2 for stochastic gradient flow

Generalization bound for NN trained by gradient flow

Compare with previous generalization bounds:



Generalization bound for NN trained by gradient flow

Compare with previous NTK-based bounds

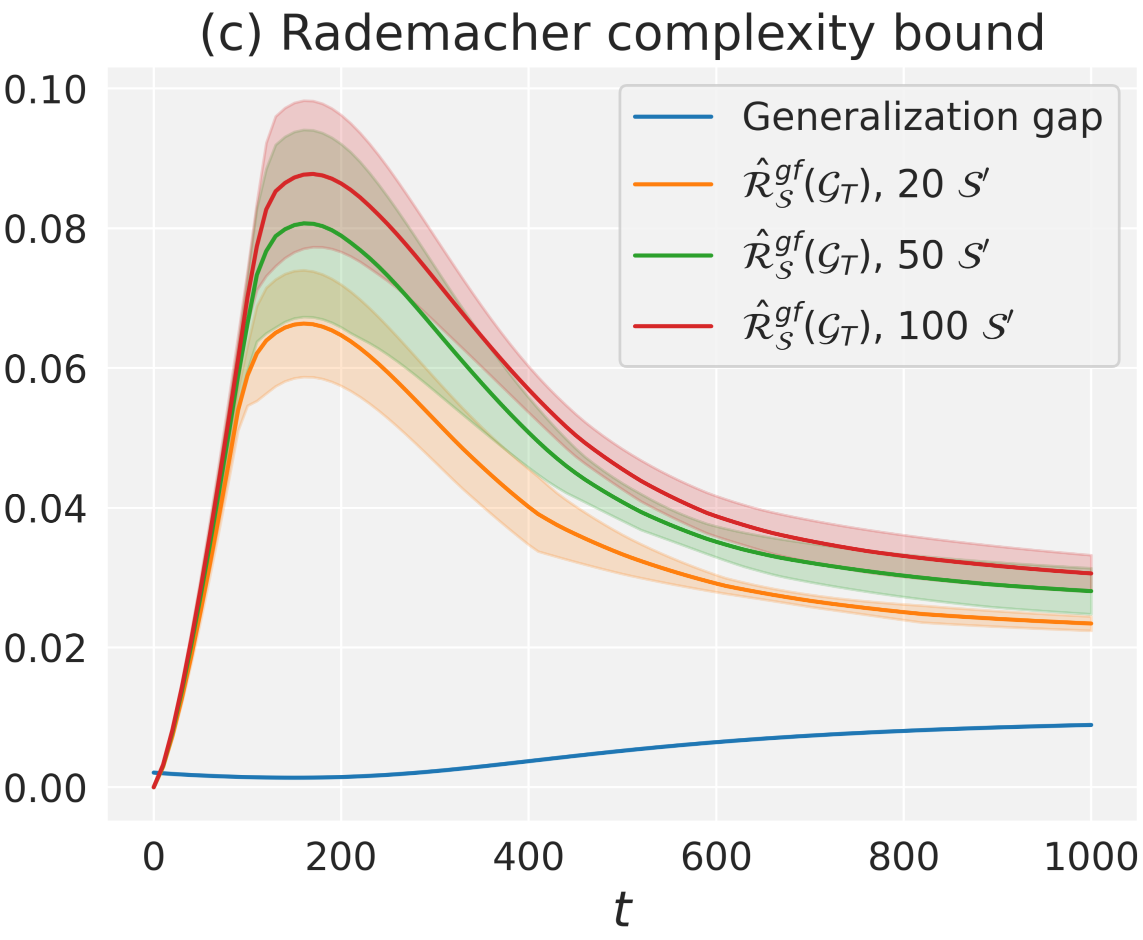
	Arora et al.	Cao & Gu	**Ours**
Bound	$\sqrt{\frac{2\mathbf{Y}^\top(\mathbf{H}^\infty)^{-1}\mathbf{Y}}{n}}$	$\tilde{O}(L \cdot \sqrt{\frac{\mathbf{Y}^\top(\mathbf{\Theta})^{-1}\mathbf{Y}}{n}})$	Theorem 3, Theorem 5
Model	Ultra-wide two-layer FCNN	Ultra-wide FCNN	**General continuously differentiable NN**
Data	i.i.d. data with $\|\boldsymbol{x}\| = 1$	i.i.d. data with $\|\boldsymbol{x}\| = 1$	i.i.d. data
Loss	Square loss	Logistic loss	**Continuously differentiable & bounded loss**
During training	No	No	**Yes**
Multi-outputs	No	No	**Yes**
Training algorithm	GD	SGD	(Stochastic) gradient flow

Much more general results!

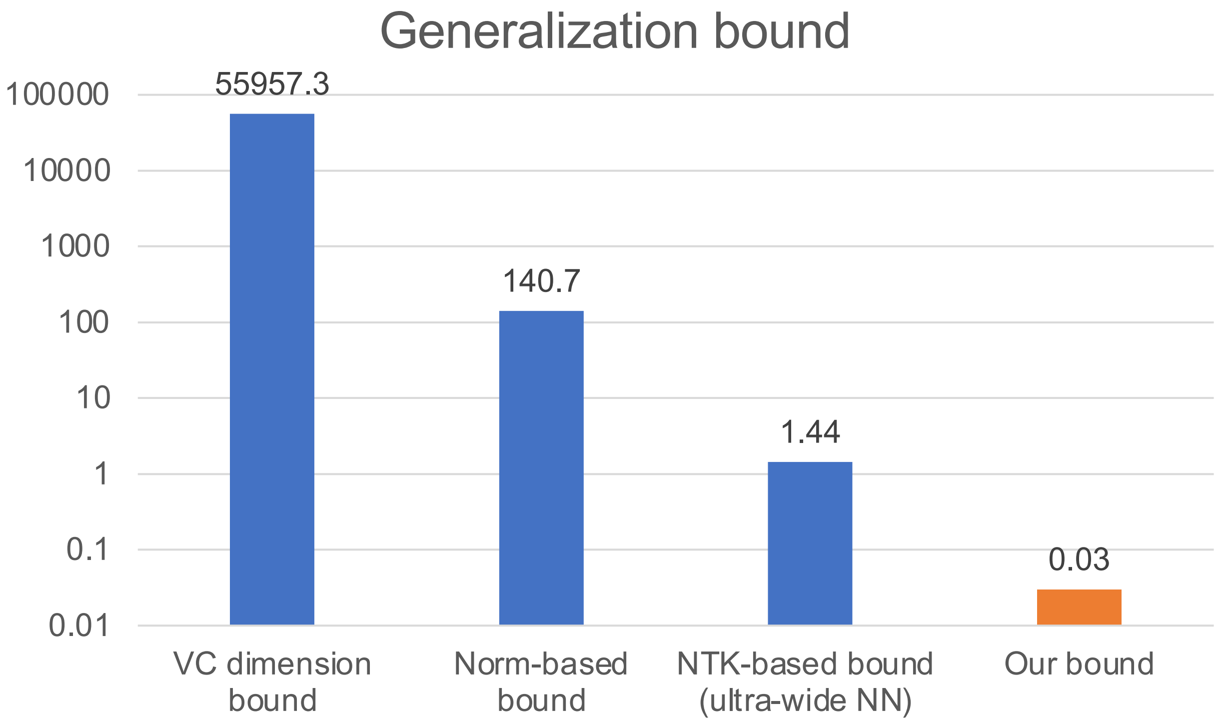
Arora et al.. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. ICML 2020
Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. NeurIPS 2019.

Experiment: Generalization bound for NN trained by gradient flow

Experiment of two-layer NN



Tight bound!



Case study: Ultra-wide NN

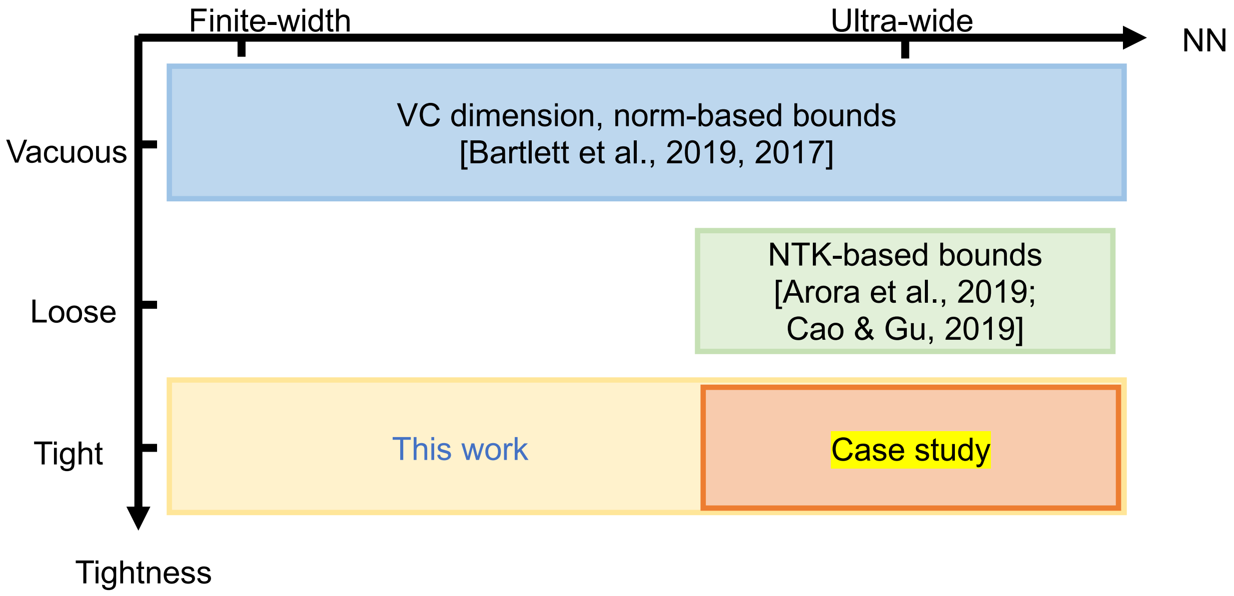
For an infinite-width NN with constant NTK $\Theta(x, x')$, we can bound LPK with NTK

$$GAP \leq \frac{\rho B \sqrt{T}}{n} \sqrt{\sum_{i,j} |\Theta(x_i, x_j)|}$$

ρ: Lipschitz constant of $\ell(f, y)$

Compare with $\tilde{O}(L \cdot \sqrt{\frac{2\, \mathbf{y}^\top (\Theta)^{-1} \mathbf{y}}{n}})$ [Cao & Gu, 2019],

1. no dependence on the number of layers L
2. holds for NNs with multiple outputs



Application: Neural architecture search

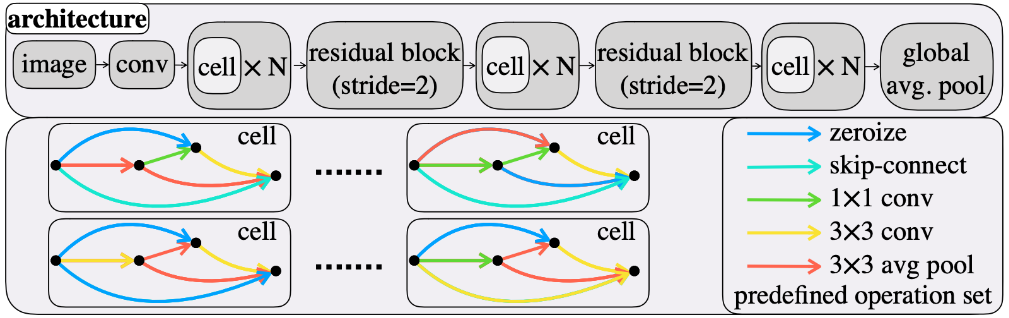
Use the bound to estimate the test loss and design minimum-training NAS algorithms:

$$\mathrm{Gene}(w,S) = L_S(w) + 2U_{sgd}$$

U_{sgd}: simplified from the bound of stochastic gradient flow

Algorithm	CIFAR-10 Accuracy	Best	CIFAR-100 Accuracy	Best
Baselines				
TENAS [13]	93.08±0.15	93.25	70.37±2.40	**73.16**
RS + LGA₃ [39]	93.64		69.77	
Ours				
RS + $\mathrm{Gene}(w,S)_1$	93.68±0.12	93.84	72.02±1.43	73.15
RS + $\mathrm{Gene}(w,S)_2$	**93.79**±0.18	**94.02**	**72.76**±0.33	73.15
Optimal	94.37		73.51	

NAS-Bench-201



"RS": randomly sample 100 architectures and select the one with the best metric value

$\mathrm{Gene}(w,S)_1$: $\mathrm{Gene}(w,S)$ at epoch 1

"Optimal": the best test accuracy achievable in NAS-Bench-201 search space

"Best": best accuracy over the four runs

Outline

1. Introduction and motivation
   - Kernel machine and neural tangent kernel
   - Generalization theory of neural networks
   - Motivation of this work
2. Main results
   - Loss path kernel and the equivalence between NN and KM
   - Generalization bound for NN trained by (stochastic) gradient flow
   - Case study and Application
     - Ultra-wide NN
     - Neural architecture search
3. Conclusion and future works

Conclusion

1 New equivalence between NN and KM

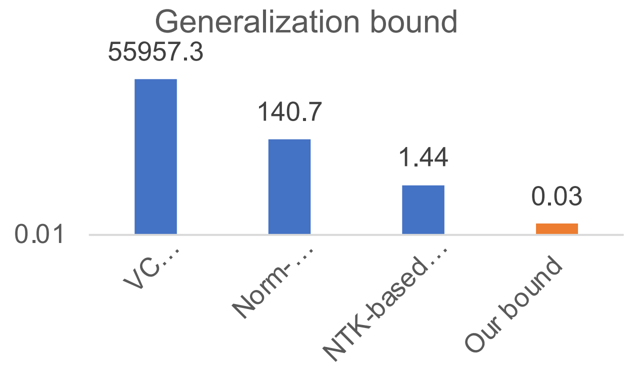
- New kernel LPK
- Much more general equivalence

$$\ell(w_T, z) = \boxed{\sum_{i=1}^{n} -\frac{1}{n} \mathrm{K}_T(z, z_i; S)} + \ell(w_0, z)$$

Kernel machine with **LPK**

2 Generalization bound for NN

- Holds for general NNs
- Tighter bounds!



3 Useful in theory and practice

- Better bound for ultra-wide NNs
- Minimum-training NAS algorithms

Algorithm	CIFAR-10 Accuracy	Best	CIFAR-100 Accuracy	Best
Baselines				
TENAS [13]	93.08±0.15	93.25	70.37±2.40	**73.16**
RS + LGA$_3$ [39]	93.64		69.77	
Ours				
RS + Gene$(\boldsymbol{w}, \mathcal{S})_1$	93.68±0.12	93.84	72.02±1.43	73.15
RS + Gene$(\boldsymbol{w}, \mathcal{S})_2$	**93.79±0.18**	**94.02**	**72.76±0.33**	73.15
Optimal	94.37		73.51	

Future works

1 Applications based on the theory

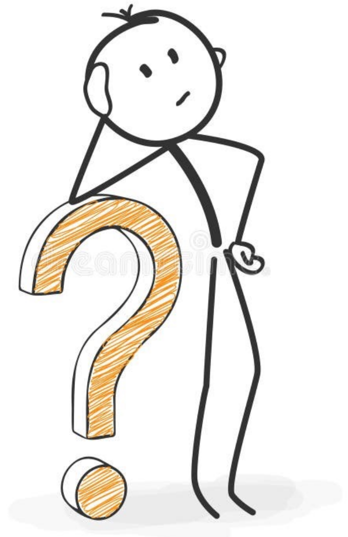
- Analyze different NN architectures from this equivalence
- Design better kernel function based on LPK
- Quantify the influence of each training sample (core set selection, interpretability, robustness)
- …

2 Equivalence & generalization for other optimization algorithms

- SGD with momentum
- Adam

3 Improve the generalization bound

- Remove supremum
- Tighter bound



References

1. Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.
2. Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. Advances in Neural Information Processing Systems, 32, 2019.
3. Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. Advances in neural information processing systems, 32, 2019.
4. Chen, Y., Huang, W., Nguyen, L., and Weng, T.-W. On the equivalence between neural network and support vector machine. Advances in Neural Information Processing Systems, 34, 2021.
5. Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences, 116(32):15849–15854, 2019.
6. Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. The Journal of Machine Learning Research, 2019.
7. Arora, S., Du, S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In International Conference on Machine Learning, pp. 322–332. PMLR, 2019.
8. Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. Advances in neural information processing systems, 32, 2019.
9. Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research, 2002.
10. Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. Spectrally-normalized margin bounds for neural networks. Advances in neural information processing systems, 30, 2017.

Thank you for listening!